JOURNAL OF
EMERGING INVESTIGATORS

Article

Comparative study on three machine learning models in novel autonomous drone-based detection of invasive plant *Brassica nigra*

Chloe Ho[1], Alex Liu[2], Yashvee Shah[3], Aarushi Shukla[4], Sahiti Pantangi[5], Dr. Larry McMahan[6]
[1] ASDRP, Basis Independent Silicon Valley, San Jose, California, USA
[2] ASDRP, Santa Clara High School, Santa Clara, California, USA
[3] ASDRP, Santa Clara High School, Santa Clara, California, USA
[4] ASDRP, Los Altos High School, Los Altos, California, USA
[5] ASDRP, Washington High School, Fremont, California, USA
[6] Chair, Computer Science and Engineering Department, ASDRP, Fremont, California, USA

SUMMARY
Invasive weeds cost farmers in the USA $33 billion a year to manage. *Brassica nigra* is an invasive annual herb which increases wildfire risk and produces chemicals that prevent the germination of native plants. We propose a solution to automate the detection of invasive plant species by creating a machine learning model capable of identifying the presence of Brassica nigra from autonomous drone footage. We tested three different machine learning models for the detection of the invasive plant: Convolutional Neural Network (CNN), Stochastic Gradient Descent Classifier (SGDC), and eXtreme Gradient Boosting (XGBoost). We hypothesized that for the detection of invasive plant species from aerial autonomous drone images, a CNN model would outperform SGDC and XGBoost because of its ability to extract spatial features to find complex visual patterns. Additionally, we hypothesized that SGDC would perform better than XGBoost, as our data is linearly separable and SGDC has the ability to do limited feature extraction. Results analyzed through heat maps of each model indicate that there is a statistically significant difference between the ability of the three models to find important features with the ANOVA test (p=9.2e^{-16}). We can conclude that CNNs are the most suitable model for detecting invasive plants from drone footage due to its superior feature extraction abilities.

INTRODUCTION
Invasive plant species are a major cause of biodiversity loss, and plants like the Ice plant have already invaded around 2 square kilometers of space in Santa Barbara, California (1,2). The California Invasive Plant Council classifies an invasive plant as a plant not native to its environment, which reproduces quickly and harms surrounding organisms, economy, and humans (3). In September 2023, over 37,000 non-native species were classified in biomes where they are not usually found. Of those species, at least 3,500 pose threats to local economy, food, and ecology (4). In fact, Center for Invasive Species research estimates that invasive weeds alone caused California $89 million in expenditures as of 2009 (5).

Over the years, a plethora of solutions have been developed to eliminate invasive plants. Integrated pest management, chemical herbicides, and physical methods like pulling and mowing have all been implemented. These solutions all depend on first finding the plant. Early Detection and Rapid Response (EDRR) is the most cost-effective way to prevent the proliferation of non-native plants (6). Drone identification is a promising prospect for EDRR because they can cover large areas efficiently and capture imagery on terrain that is difficult to survey manually. For example, a prior study mapped invasive plant growth patterns via drone imagery, demonstrating the feasibility of drone surveys while requiring manual control of the drone and ground reference marks(7). However, current literature contains several inefficiencies including the necessity of ground markers as a reference point and manual control of the drone as it flies along a pre-programmed flight path.

To address the current limitations, we created an autonomous drone that flies along a pre-programmed flight path and collects footage that is analyzed with machine learning to identify the presence of the invasive plant *Brassica nigra*. *B. nigra*, or the black mustard plant, is an invasive plant that threatens Bay Area ecology as it competes with native plants, increases fire risks, and produces allelopathic chemicals which affect native plant growth (8). *B. nigra* is identified by its yellow flower, ovular shape, orientation, and stem and leaf growth patterns. We selected *B. nigra* as the focus species of this study because of its accessibility and ability to quickly reproduce. *B. nigra* is highly abundant in the Bay Area and grows across various regions that can be surveyed by drones. Additionally, its distinct features assist effective image classification.

To detect *B. nigra* from drone footage, we tested three different machine learning techniques: Convolutional Neural Network (CNN), Stochastic Gradient Descent Classifier (SGDC), and eXtreme Gradient Boosting (XGBoost). CNNs have strong pattern recognition abilities and learn spatial features because of the math computation based on clusters of pixels (9). A CNN model finds parts of the image which are most useful for classification and gives more weight to those pixels during the mathematical computation. Because of this, CNNs are able to recognize flower petal shape, color distribution, and more patterns allowing it to better determine whether the plant in the image is invasive or not. SGDC is a linear model and works well if the data is linearly separable. It is efficient on large datasets, performs well on simple, well-separated patterns, and is strong in situations where feature interpretability, or interpreting which features parts of the image are most influential in model predictions, matters

https://doi.org/10.59720/25-115

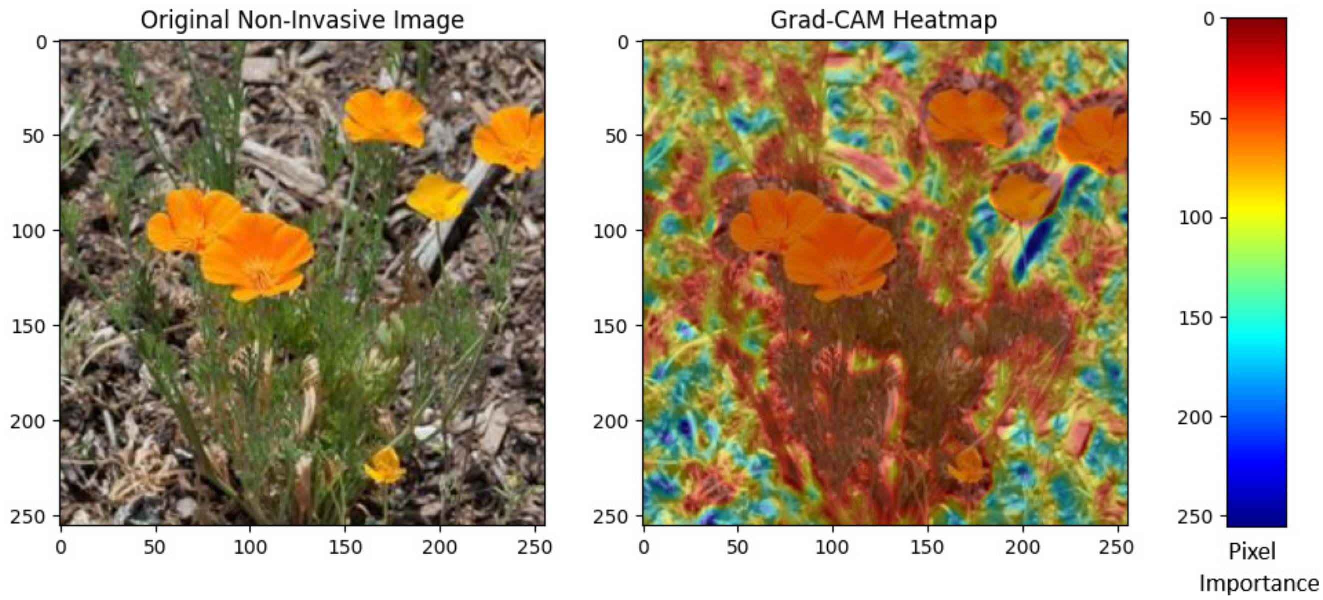

Figure 1. CNN heat map for Non Invasive Plant Image. Visualization of CNN using Grad-CAM. The image on the left shows the origial input photograph of the non-invasive flower and the image on the right displays a Grad-CAM (Graident-weighted Class Activation Mapping) heat map. The heat map shows the regions of the image that most strongly influenced the CNN's classification decision, with warmer colors indicated higher importance. The heat map focuses on the flower, demonstrating that the model focuses on the flower when making its prediction and is able to distinguish the flower from the environment..

(10). XGBoost is a tree-based model. In the case of binary classification of native and invasive plants, SGDC does not capture complex spatial patterns of the invasive species as adequately as the CNN due to limited recognition of spatial features. XGBoost handles mixed types of features efficiently and is naturally robust to outliers and noisy features through its boosting process (11). However, XGBoost is less effective inadequate with data that contain raw image pixels and is better with structured data such as numerical data (leaf size, petal size, number of petals, etc.) (12). We chose these models because they are all based on different computational methods: pixel computation, linear regression, and tree-based. Due to these factors, we hypothesized that a CNN model will outperform SGDC and XGBoost when detecting invasive plant species from aerial autonomous drone images because of its ability to extract spatial features to find complex visual patterns (13). Additionally, we hypothesized that SGDC will perform more accurately than XGBoost, as our data is linearly separable and SGDC has the ability to do limited feature extraction. This is necessary because it proves that the model can focus on key data points and noise reduction to come up with a higher accuracy than the XGBoost.

Our results showed that the CNN model achieved the highest accuracy for classification and was able to extract spatially significant features from invasive plant images, outperforming the other two models. These findings suggest how image classification using a CNN for drone identification holds strong potential for the detection of *B. nigra*, providing land managers with advantageous data to guide mitigation efforts.

	Accuracy	F1 Score	Precision	Recall
CNN	99.4%	99.4%	99.1%	99.7%
SGDC	95.3%	95.2%	92.7%	97.8%
XGBoost	90.8%	90.7%	90.5%	91.1%

Table 1: Model Prediction Results for all Three Models. Accuracy, f1, precision, and recall score when classifying plants as native or invasive using the convolutional neural network (CNN), stochastic gradient descent classifier (SGDC), and eXtreme gradient boosting (XGBoost). Models were sorted from highest accuracy to lowest accuracy.

RESULTS

In order to compare the performance of the three different models, we evaluated accuracy, feature extraction, and statistical significance between the models. As predicted, the CNN model performed exceptionally and demonstrated optimal ability to recognize important features of an image, outperforming both SGDC and XGBoost. It achieved the highest accuracy of 99.4%, F1 score of 99.4%, precision score of 99.1%, and recall score of 99.7% (**Table 1**). The accuracy was calculated as the proportion of correctly classified images relative to the total number of images in the test dataset. indicating that the CNN effectively classified images of the invasive species. The CNN model effectively separated the flower and leaves in each image, regardless of the location of the flower (**Figure 1**). When we converted the heat map into pixel values and the sum of all rows and columns are plotted, there was substantiative variance and a high standard deviation as the Y pixels has a wide range, indicating that some rows and columns are more important than others and that the CNN can determine these rows and columns by weighting them more strongly (**Figure 2**).

SGDC and XGBoost were less effective in comparison to

https://doi.org/10.59720/25-115

Figure 2. CNN Heat Map Sum Plot. The sums of the CNN Grad-CAM heat map values across rows and across columns. Summing the heat map values along each axis shows how strongly different horizontal and vertical regions of the image contribute to the model's prediction. Larger variance in sums indicates the model's attention is concentrated in specific regions of the image, demonstrating the CNN is distinguishing the plant from the environment.

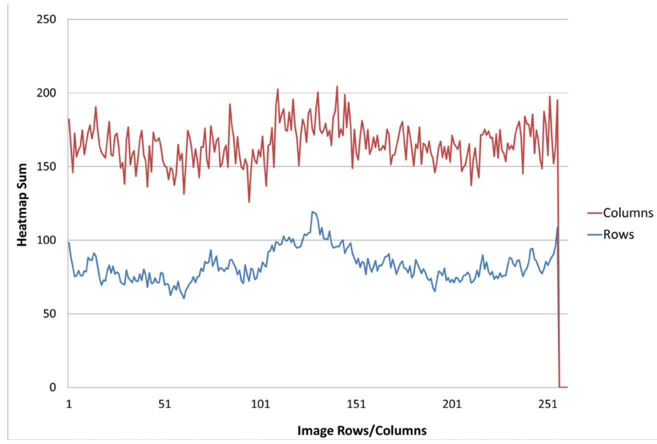

Figure 4. SGDC Heat Map Sum Plot. The sums of the SGDC heat map values across rows and across columns. Summing the heat map values along each axis shows how strongly different horizontal and vertical regions of the image contribute to the model's prediction. Larger variance in sums indicates the model's attention is concentrated in specific regions of the image.

the CNN model. SGDC achieves the second-best accuracy of 95.3%, F1 score of 95.2%, precision score of 92.7%, and recall score of 97.8% (**Table 1**). SGDC is capable of recognizing important features and patterns (**Figure 3**), but the heat map sum plot demonstrated weaker performance than a CNN model (**Figure 4**). The SGDC heat map values of the sum of rows and columns had a smaller standard deviation than the CNN values showing that it is not as good at determining important features of the image, valuing each row and column more similarly to the other rows and columns, while still performing better than XGBoost.

XGBoost had the lowest accuracy of 90.8%, F1 score of 90.7%, precision score of 90.5%, and recall score of 91.1% (**Table 1**). It also demonstrated the lowest standard deviation of summed pixel intensities across rows and columns and

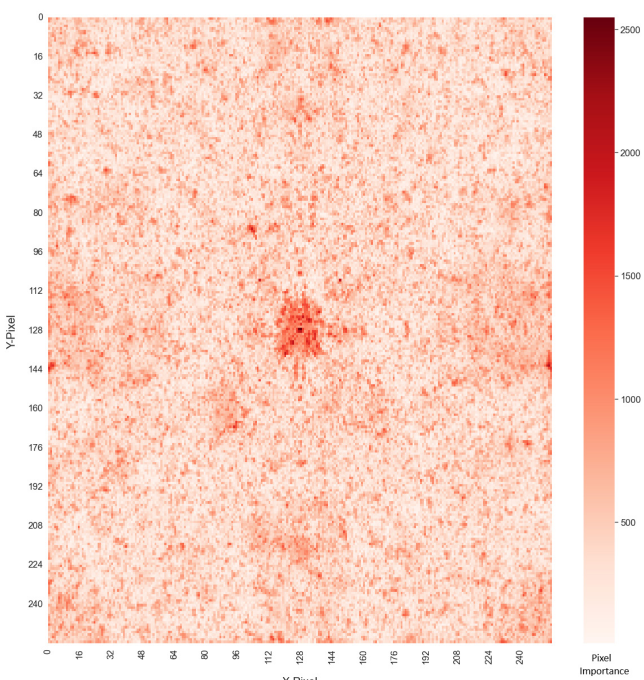

Figure 3. SGDC Heat Map. The heat map shows the relative influence of each pixel on the SGDC model's classification. Darker red pixels indicate regions that contribute more strongly to the model's prediction, while lighter regions have less influence.

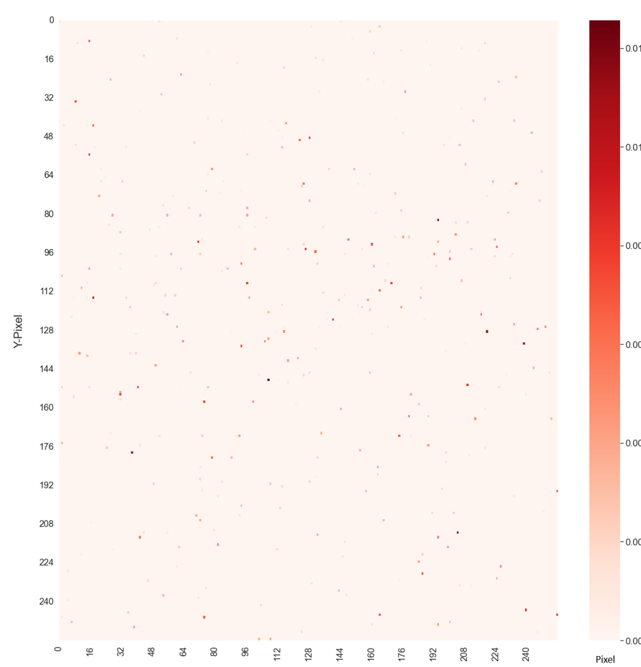

Figure 5. XGBoost Heat Map. The heat map shows the relative influence of each pixel on the SGDC model's classification. Darker red pixels indicate regions that contribute more strongly to the model's prediction, while lighter regions have less influence.

https://doi.org/10.59720/25-115

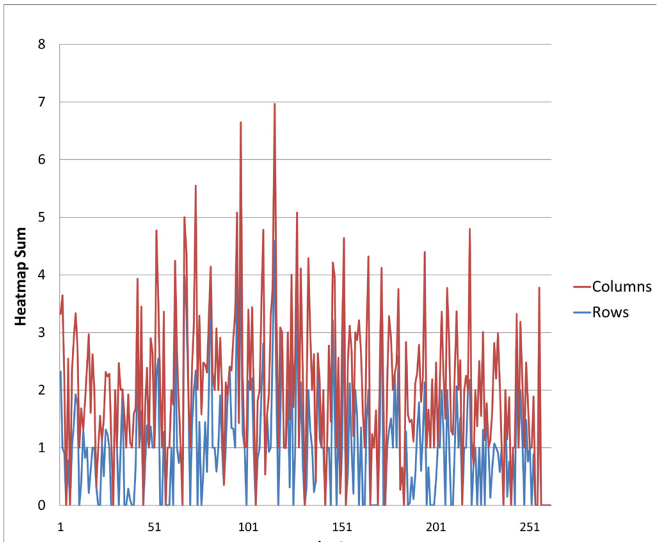

Figure 6. XGBoost Heat Map Sum Plot. The sums of the XGBoost heat map values across rows and across columns. Summing the heat map values along each axis shows how strongly different horizontal and vertical regions of the image contribute to the model's prediction. Smaller variance in sums indicates the model's attention is not concentrated in specific regions of the image.

required longer training time as it is a tree-based model. The XGBoost heat map showed that this model lacks the ability to recognize important features of the images as the plot is steadily distributed (**Figure 5**). The XGB heat map sum plot's y-axis has a very small range, indicating that there is little variance between the sums of each row and column, meaning the model could not determine important parts of the image (**Figure 6**). These results are unlike those of SGDC and CNN, which can identify patterns and recognize what is important, abd XGBoost also cannot learn the spatial features.

Each model was run five times to achieve 5 standard deviations of the row and column pixel intensity and to determine if those differing standard deviations were indeed statistically significant. The mean summed standard deviation values across the five runs were 1.31 ± 0.36 for the XGBoost, 20.90 ± 0.45 for the SGDC, and 36.14 ± 1.14 for the CNN. Running an ANOVA test on these values and calculating the F-statistic and p-value with an alpha level of 5% revealed that the differences in the standard deviation metric across the three models were statistically significant ($p = 9.2e^{-16}$). Based on these results, the CNN model was better than the other two in detecting the invasive species *B. nigra* from autonomous drone aerial images.

DISCUSSION

Autonomous drone-based detection of invasive species represents a promising approach for EDRR, yet prior work has relied on manual drone operation and the use of ground-based reference markers. This study evaluated if machine learning models trained on aerial imagery taken by an autnomous drone could differentiate *B. nigra* from native species. With high standard deviation between the heat map values of each row and column that demonstrate good pattern recognition abilities and statistically significant results, our results confirm that the CNN model is more suitable for extracting important features of images with *B. nigra* and predicting if the feature exists in the test data than SGDC and XGBoost.

In the future, it would be interesting to expand our model to detect other invasive species including *B. nigra* or recognize specific native species that surround the detected invasive species. This could extend our approach to detect species that include fungi, trees, or other flowers. We could also work to extend the drone's battery life to increase the time of autonomous flight and collect more data at once. The practical applications of this project can be expanded by partnering with local national parks, compiling a list of their invasive species, and establishing a long-term system to utilize our drone to track the proliferation and population fluctuations of the species in the park.

However, there are some limitations in the quality of the training and testing data. Because of drone flight restrictions, we could not fly and test our drone in more plant dense areas, such as nearby national parks. The seasonal nature of flowers also meant that much of the data recorded during winter months contained limited examples of *B. nigra*. However, seasonal variations across our drone footage also suggest that our model is usable for a variety of seasons. The data could be improved by having more types of flowers both when they are alive and withering to further ensure that the model does not misclassify other flowers as *B. nigra.*

To conclude, our project provides evidence that CNN models outperform SGDC and XGBoost models in discerning features from drone footage and accurately classifies the resulting images based on the presence of the *B. nigra* plant. We propose that a CNN model is well suited for binary classification of invasive and native species and that an autonomous drone allows for the efficient collection of data. This can help modern day conservation efforts as local authorities can use this model to efficiently find and eliminate harmful invasive species.

MATERIALS AND METHODS

Configuring Drone Prior to Flight

For drone assembly, 3D-printed parts were used to mount the flight controller (Pixhawk 4), GPS and compass (Holybro M10 GPS), ELRS radio receiver (RadioMaster RP3 ELRS FPV), ELRS radio transmit (Radio Master Boxer), battery (Lipo 5000mhA), two rangefinders (Benewake TFmini plus lidar), and camera (GoPro 11) securely onto the drone frame (Holybro X500 V2 ARF). Holybro provided a 3D model of their drone frame for reference. The hardware components were wired and connected following instructions for connecting the UARTs, I2Cs, motor, and power.

In this project, ArduPilot was used as the flight software stack. ArduPilot is an open-source autopilot software that is commonly used for autonomous or semi-autonomous vehicle configuration. Mission Planner, a ground station software (GSC) that enables communication between a computer and the drone's flight controller, was used to configure the drone. To fly the custom-built drone, the power module, receiver, and GoPro camera with the GPS and compass were configured following ArduPilot's website and tuned before it can fly stably and safely. This involved following the steps outlined by the ArduPilot tuning procedures (14,15).

https://doi.org/10.59720/25-115

To configure the drone to fly autonomously, a flight plan was set up (16). In this project, we set up flight plans using both Mission Planner and QGroundControl. By configuring a survey grid and downloading it to the drone, the drone was able to survey an area and capture photos with the camera. For autonomous flight, the drone needed to be armed and switched to the AltHold mode, a fight mode that keeps the drone at a constant altitude. Once the drone was stable, switching it to the Auto mode allowed the drone to survey the field based on the survey grid from the GCS. For this project, the drone was flown at 2.5 meters per second at a height of 7 meters.

Machine Learning Models

For this project, a convolutional neural network, SGDC, and XGBoost were trained to classify whether the aerial images were native or invasive species. As there are only two classes, binary classification was used.

The images had various resolutions and color channels because the images were taken with different cameras and settings. As such, all images were converted into JPEGs with 3 color channels (RGB), cropped into a square (256 x 256), and labeled invasive or native. Each image is then represented as a matrix that contains the pixel intensity values of each pixel of the image. These pixel values form the numerical input used by the models. With data augmentation, 1300 native images and 330 invasive images were rotated, zoomed, and flipped to make a total of 100,000 native and 100,000 invasive images, creating a more balanced dataset with an equal number of invasive and native images (17). 80% (160,000 images) of the data was used for training and 20% (40,000 images) of the data was used for testing. Half the images of each set were invasive and the other were native. For image loading, a custom image loader was written to process the CSV file that contained the image and its class label. Images were loaded and processed in a batches of 64 images (18).

For this project, we primarily relied on default hyperparameter settings from the documentation for each model, rather than conducting an exhaustive hyperparameter search. This choice was made to ensure stability and reproducibility of results within the scope of the project, since the primary goal was to compare the relative performance of CNN, SGDC, and XGBoost rather than to maximize each model individually.

After all images was converted to numerical pixel values, the data was normalized using the mean values of 0.4914, 0.4822, and 0.4465 and standard deviations of 0.2023, 0.1994, and 0.2010 for each R, G, and B channel respectively (18).These values correspond to the average pixel intensity and variation and are normalization parameters used to rescale the image data to improve model training. The CNN model was written using Pytorch and had 4 convolutions, 2 max poolings, a linear conversion layer, and a final sigmoid activation layer (18) (**Figure 7**). The loss function was binary cross entropy. The SGD optimizer was used with a learning rate of 0.001, weight decay of 0.005, and momentum of 0.9 (18,19). The epoch loss was 0.007 with 10 epochs.

Using the Python Sklearn library's SGDClassifier, a SGDC model, a linear classification model, was trained using the fit function on the 160,000 training images (20). L2 normalization was applied to the data and the data was flattened into a 1D vector. It took 206.82 seconds to train and used log loss.

Using the Python Sklearn library's XGBoost model, XGBoost was trained using the fit function on the 160,000 training images (21). L2 normalization was applied to the data and the data was flattened into a 1D vector. It took around 4 hours to train. This is because XGBoost is a tree classifier and due to all the branches used, it takes a long time to train.

Model Testing and Analysis

For testing, each model was run on the testing data and predicted what class each image belonged to. If the predicted class was the same as the label, then the model predicted correctly. Accuracy, F1, precision, and recall scores were also calculated based on the model's predictions. If the model predicted that the image contains an invasive species, then the GPS location was extracted from the GoPro image and saved to aid with the removal of the invasive plant.

For the CNN model, a heat map was created to determine which features are important for prediction (22). For the SGDC and XGBoost models, feature importance plots were created to show which pixels the model considered most important for prediction. These plots were done on the same image for each model. To analyze these plots, we examined the intensity of heat map. Analyzing each individual pixel is inefficient, so we summed the values of each row and column in the image and computed the standard deviation of the rows and columns. Summing the two values (standard deviation of the row sums and standard deviation of the column sums) provides a metric that can be used to compare each model and determine which one is better at recognizing important

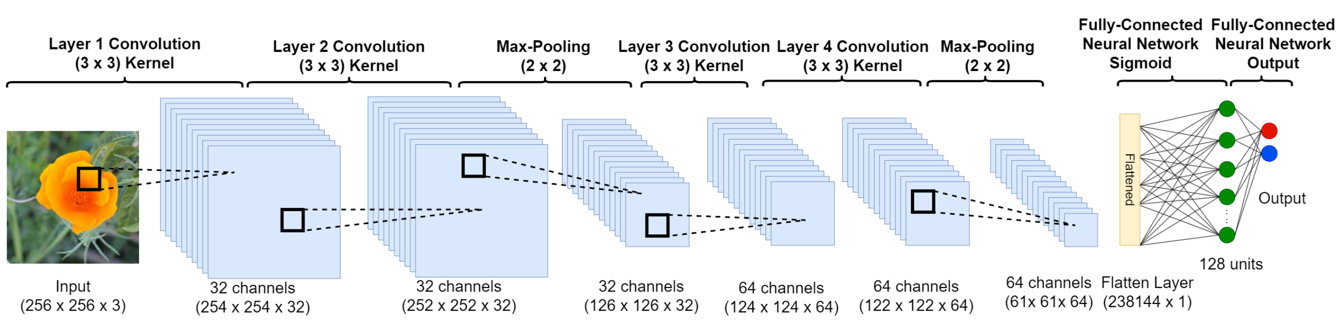

Figure 7. CNN model. Diagram of the convolutional neural network used in this project, showing each layer and how the model processes the input image to produce the final classification output.

features. If the sums of the rows and columns of the plots have small standard deviations from each other, then the model cannot distinguish which pixels are more important as each row and column are valued similarly to the others, but a large standard deviation indicates that the model can determine which parts of the image are more important. We then ran an ANOVA test to determine if the summed standard deviations of the three models are significantly different from each other.

Received: March 3, 2025
Accepted: November 13, 2025
Published: July 05, 2026

REFERENCES

1. Rai, Prabhat and Singh, JS. "Invasive alien plant species: Their impact on environment, ecosystem services and human health." *Ecological Indicators*, vol. 111, 9 Jan 2020. https://doi.org/10.1016/j.ecolind.2019.106020. Accessed 15 Aug. 2025.
2. Garcia et al. "Mapping invasive iceplant extent in southern coastal California using high-resolution aerial imagery." *Ecological Informatics*, vol. 81, July 2024. https://doi.org/10.1016/j.ecoinf.2024.102559. Accessed 15 Aug. 2025.
3. "About invasive plants." Cal-IPC, https://www.cal-ipc.org/plants/impact/. Accessed 15 Aug. 2025
4. Mamer, Eric. "Biodiversity: new IPBES report finds invasive alien species a growing and costly threat worldwide." EU Environment, 4 Sept. 2023. https://environment.ec.europa.eu/news/biodiversity-new-ipbes-report-finds-invasive-alien-species-growing-and-costly-threat-worldwide-2023-09-04_en. Accessed 15 Aug. 2025.
5. "The Cost of Invasive Plants in California." Cal-IPC, https://www.cal-ipc.org/solutions/research/cost/. Accessed 15 Aug. 2025
6. "Invasive Species Control | U.S. Department of the Interior." U.S. Department of the Interior, 15 Mar. 2017. https://www.doi.gov/ocl/invasive-species-control. Accessed 21 Apr. 2025.
7. Boise State University. "Detecting Invasive Species Using Drones." Youtube, 2022. https://www.youtube.com/watch?v=b0OcuvfI19w&ab_channel=BoiseStateUniversity. Accessed 15 Aug. 2025.
8. "Brassica Nigra Profile." California Invasive Plant Council. https://www.cal-ipc.org/plants/profile/brassica-nigra-profile/. Accessed 21 Apr. 2025.
9. Razzak, Razia, *et al*. "Vulgar Comments Classification: Comparison between CNN, XGBoost and SVM." *Research Square*, 9 Aug. 2023. https://doi.org/10.21203/rs.3.rs-3185443/v1.
10. "ML - Stochastic Gradient Descent (SGD)." GeeksforGeeks, https://www.geeksforgeeks.org/machine-learning/ml-stochastic-gradient-descent-sgd/. Accessed 16 Aug. 2025.
11. "XGBoost." Nvidia, https://www.nvidia.com/en-us/glossary/xgboost/. Accessed 16 Aug. 2025.
12. Shwartz-Ziv, Ravid, and Amitai Armon. "Tabular Data: Deep Learning Is Not All You Need." *Information Fusion*, vol. 81, May 2022, pp. 84–90. https://doi.org/10.1016/j.inffus.2021.11.011.
13. Qin, Pengda, *et al*. "An Empirical Convolutional Neural Network Approach for Semantic Relation Classification." *Neurocomputing*, vol. 190, 19 May 2016, pp. 1–9. https://doi.org/10.1016/j.neucom.2015.12.091.
14. "First Flight With Copter." ArduPilot,. https://ardupilot.org/copter/docs/flying-arducopter.html. Accessed 21 Apr. 2025.
15. "First Time Setup." ArduPilot. https://ardupilot.org/copter/docs/initial-setup.html. Accessed 21 Apr. 2025.
16. "Mission Planning." ArduPilot. https://ardupilot.org/copter/docs/common-mission-planning.html. Accessed 21 Apr. 2025.
17. "Python | Data Augmentation." Geeks for Geeks, 3 Aug. 2023. https://www.geeksforgeeks.org/machine-learning/python-data-augmentation/. Accessed 21 Apr. 2025.
18. Nouman. "Writing CNNs from Scratch in PyTorch." DigitalOcean, 16 Sept. 2024. www.digitalocean.com/community/tutorials/writing-cnns-from-scratch-in-pytorch. Accessed 21 Apr. 2025.
19. "SGD." PyTorch 2.6 Documentation. https://docs.pytorch.org/docs/stable/generated/torch.optim.SGD.html. Accessed 21 Apr. 2025.
20. Pedregosa et al. "Scikit-learn: Machine Learning in Python." Journal of Machine Learning Research, vol. 12, 2011, pp. 2825-2830. https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html.
21. "XGBoost Documentation." XGBoost. https://xgboost.readthedocs.io/en/release_3.0.0/index.html. Accessed 21 Apr. 2025.
22. Waskom, Michael. "seaborn.heatmap." Seaborn. https://seaborn.pydata.org/generated/seaborn.heatmap.html. Accessed 21 Apr. 2025.